Аннотация

В работе ставится задача извлечения пользовательских атрибутов из диалоговых реплик в структурированном формате, для дальнейшего использования их в персонализованных диалоговых агентах или в рекомендательных системах. Из-за отсутствия данных с качественной разметкой, используется техника разметки с помощью моделей из других задач, добавляющая шум в данные. Предлагаются два способа решения поставленной задачи на основе трансформеров: решение с одной моделью (GenAE) и с двумя моделями (PipelineAE). Обсуждаются недостатки и преимущества обоих подходов. По итогу проведенных экспериментов получены результаты выше чем у существующих подходов решения данной задачи. Предложены методы улучшения качества разработанного framework'а и направления для дальнейших исследований.

Содержание

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящем отчете о НИР применяют следующие сокращения и обозначения:

Авторегрессионная генеративная модель — модель генерирующая следующий элемент входной последовательности, неявно используя условное распределение следующего элемента по предыдущим.

Метод k-ближайших соседей (англ.: k-nearest neighbors algorithm, kNN) — алгоритм для автоматической классификации объектов или регрессии.

Персона — описание личности в разговорных диалоговых датасетах. Обычно состоит из 5-6 коротких предложений о пользователе и его предпочтениях.

Трансформер (англ.: Transformer) — архитектура глубоких нейронных сетей, использующая механизм внимания для повышения скорости обучения.

Триплет — кортеж из трех элементов: (субъект, отношение, объект). Отношение или предикат определяется из конечного множества.

Batch — подмножество объектов из всего датасета, которое обычно выбирается случайно. Используется в обучении современных больших нейронных сетей с помощью стохастического градиентного спуска или Adam.

BERT (сокращ.: Bidirectional Encoder Representations from Transformers, рус.: двунаправленные векторные представления трансформера) — архитектура и метод предобучения трансформера, при котором модель должна предсказать токены последовательности, замененные на специальный MASK-токен, а также, является ли второе предложение из двух поданных на вход продолжением первого в тексте.

Checkpoint — промежуточные сохраненные веса модели во время обучения для фиксации состояния обучения.

Collaborative filtering — подход к построению рекомендательных систем, при котором контент/товары рекомендуются на основании схожести пользователей.

Content-based filtering — подход к построению рекомендательных систем, при котором пользователи сопоставляются с контентом или товарами, которые им нравятся.

Distant supervision — прием использующийся в построении графов знаний. Основная идея и предпорложение приема заключается в том, что если какие то две сущности находятся в определенном отношении, то любое предложение содержащее эти сущности выражает это отношение.

Encoder-decoder — архитектура моделей которые решают задачи sequence-to-sequence. Encoder получает некоторые представление входных данных, a decoder преобразовывает это представление в выходные данные.

Few-shot — разновидность обучения с учителем, когда обучающих примеров чрезвычайно мало: обычно 5 или 1.

Framework — комплексное программное обеспечение которое содержит в себе методы преобразования данных и обучение моделей вместе с методами оценки их качества.

Gated Recurrent Unit (сокращ.: GRU, рус.: управляемый рекуррентный блок) — разновидность архитектуры рекуррентных нейронных сетей, обладающая улучшенными свойствами обработки долговременных последовательностей, но при этом имеющая меньшее число параметров по сравнению с LSTM.

GBDT (Gradient Boosting Decision Tree) — алгоритм дерева решений, основанный на итеративном накоплении.

Logit — выходные данные модели до применения активации Sigmoid или Softmax.

LSTM (Long Short-term Memory, рус.: долгая краткосрочная память) — разновидность архитектуры рекуррентных нейронных сетей, обладающая улучшенными свойствами обработки долговременных последовательностей.

MLM (сокращ.: masked language modelling) — задача в обработке естественного языка, когда маскируется случайное слово во входном предложении и модель должна предсказать какое слово было замаскировано. Используется в предобучении BERT-подобных моделей.

Multi-Head Self-Attention — разновидность механизма внимания, которая дает возможность каждому входному вектору взаимодействовать с другими входными векторами.

Multi-label classification — разновидность задач обучения с учителем, когда по входному объекту предсказывается сразу некоторое множество классов, которое может быть и пустым. Например теги к фильмам в онлайн кинотеатре.

Point-wise Feed-Forward Network — разновидность сети прямого распространения, состоящая из двух слоев и решающая задачу регрессии.

Position-wise Feed-Forward Network — разновидность сети прямого распространения, состоящая из двух полносвязных слоев, применяемых к последнему измерению, что означает, что для каждого элемента последовательности используются одни и те же полносвязные слои.

Retrieval (рус.: ретрив) — процесс извлечения наиболее релевантных к входным данным элементов из данного набора кандидатов, например ответа на вопрос.

Seed — число или вектор использующийся для инициализации псевдослучайного генератора чисел.

Self-Attention — разновидность механизма внимания, выявляющая закономерности

только между входными данными.

Sequence-to-sequence — генерация нейронной сетью выходной последовательности векторов по входной последовательности, при этом каждый следующий токен выходной последовательности генерируется с использованием предыдущих сгенерированных токенов.

Seq2seq — Sequence-to-sequence модель глубокого обучения, принимающая на вход последовательность элементов, и возвращающая другую последовательность элементов.

SOTA (сокращ.: state-of-the-art) — высший уровень развития некоторой технологии, или метод который получил самый лучший результат в определенной задаче.

ВВЕДЕНИЕ

Атрибуты пользователя — это явные представления личности и характеристик человека в структурированном формате. Они предоставляют собой богатое хранилище личной информации для лучшего понимания пользователей во многих приложениях. Тем не менее, качественные пользовательские атрибуты получить сложно, поскольку информация в социальных сетях, часто распределена сильно разреженно. Таким образом, использование неструктурированных источников данных для получения структурированных пользовательских атрибутов является сложным направлением исследований.

Между тем, люди все больше полагаются на диалоговых агентов, чтобы помогать, информировать и развлекать людей, например, составлять компанию пожилым людям и обеспечивать обслуживание клиентов. Данные о разговорах между пользователями и системами информативны и многочисленны, и большинство существующих подходов в глубоком обучении создаются на основе больших данных полученных путем crowd-source'а или извлеченных разговоров. Часто, в таких подходах учитывается только текущий контекст диалога, то есть несколько предыдущих реплик, и строится ответ на основе контекста, либо дополнительно используются атрибуты самой системы для создания последующих хороших ответов. Тем не менее, вся история диалогов одного и того же человека игнорируется, что означает, что эти системы не строят знакомство с пользователями постепенно, извлекая пользовательскую информацию из разговоров.

Цель данной работы заключается в создании современного framework'а извлечения пользовательских атрибутов в достаточно универсальном структурированном формате из диалогов, для дальнейшего использования их в различных подзадачах, например в рекомендательных системах. Ставится задача предсказания данных о пользователях, которые представляются в формате кортежа: *(субъект, отношение, объект)* из данной реплики, которые в данной работает называются "триплетами". Например, в реплике "I have walked with my dog this morning." содержится кортеж *(I, have pet, dog)*. Тем не менее, стоит заметить, реплики могут содержать, либо никаких кортежей, либо сразу несколько. Например, в реплике "Good morning, how are you?" нет никакой информации о пользователе, и соответственно нет кортежа. А в предложении "I took my son to school on my black Lada yesterday." есть два кортежа: *(I, have children, son)* и *(I, have vehicle, car)*. Важно, так же отметить, что постановка задачи предполагает что framework принимает на вход только утверждения, а не вопросы. То есть если есть пара вопроса и ответа где выясняется существование некоторого атрибута пользователя, а ответ служит подтверждением или отрицанием, то данную пару необходимо преобразовать в одно утверждение от лица пользователя. Например, "Q: What pet do you

own? A: A dog.” $\Longrightarrow$ ”I have a dog.”. Пример данной задачи можно увидеть на Рисунке 1.

	Conversations	**User Attributes**
Usr	Hello, how are you doing today?	none
Sys	I am fine! Where do you live?	
Usr	I am originally from California but now I live in Florida for long.	(I, live_in, Florida)
Sys	Florida! You must have a good work-life balance.	
Usr	Oh, I no longer work at banks but for exercise I walk often.	(I, previous_profession, banker) (I, has_hobby, walking)
Sys	Good to hear that! Do you live with your family?	
Usr	My son. I bring him to church every Sunday with my Ford.	(I, has_children, son) (I, like_goto, church) (I, have_vehicle, ford)
Sys	Wow sounds good! You can meet many people.	
Usr	Sure, but my son is afraid of talking to others.	(My son, misc_attribute, shy)

Рис. 1: Реплики и извлеченные из них кортежи. Пример взят из статьи [1]

Основную сложность в данной задаче определяет отсутствие датасета с качественной разметкой и методов обучения без учителя.

ОСНОВНАЯ ЧАСТЬ

1 Релевантные исследования

К представленному в данной работе исследованию имеются несколько смежных задач и направлений. Ниже описан краткий обзор по каждому направлению и существующих подходов решения поставленных там задач.

Извлечение персональных атрибутов Большинство работ по извлечению персональных характеристик из естественного языка использовали технику distant supervision и эвристические методы и шаблоны ([2], [3], [1], [4]), у которых довольно низкая полнота (recall). В исследованиях в этой работе так же используется distant supervision. Тем не менее, он не критичен для работоспособности framework'а и служит исключительно как способ оценить качество представленных подходов и сравнить их с другими методами. В [5] представлен датасет, состоящий из повседневных диалогов с описаниями персон каждого собеседника из 5-6 предложений, и ставится задача генерации ответа по входному контексту диалога используя персону собеседника и собственную персону. Тем не менее, выделение релевантных частей персоны к определенной реплике не является основным фокусом исследования. Аналогичная работа была проделана в статье [6]: в описанной архитектуре модели используется компонента retrieve'а нужного участка персоны, которая наиболее релевантна к текущему контексту диалога. Тем не менее, эти работы не ориентируются непосредственно извлекать эти атрибуты из диалогов, а использовать их в генерации ответа неявным образом. Наиболее близкими исследованиями к данной работе являются [1] и [7]. В [1] представляется архитектура модели состоящая из двух компонент: классификатор отношений и генератор сущностей, и обе обучаются в сквозном режиме как одно целое. В одном из подходов решения задачи в данной работе так же было решено придерживаться этой архитектуры, однако позже будет показано, что эти компоненты могут быть не связаны друг с другом для достижения хорошего качества. Стоит так же отметить, что в отличие от этого подхода, подход представленный в данной работе использует модели основанные на трансформер [8], что заметно улучшает качество. Другой подход, описанный в [7] так же использует две компоненты: одну для генерации триплетов, и другую - для оценки релевантности сгенерированного триплета к реплике. Задачу извлечения признаков авторы [7] разделяют на две подзадачи: на явное извлечение атрибутов - когда некоторое отношение является подстрокой в реплике, и неявное - когда отношения нужно выводить из реплики основываясь на семантику. Хотя подход показал себя хорошо в указанных подзадачах, в общем случае, неизвестно под какую

из этих подзадач подходит входная реплика.

Построение графов знаний Установленный формат в котором извлекается информация в данной работе очень похож на структуру графов знаний, в которых граф так же хранится в виде кортежей из трех элементов: две сущности, и отношение между ними. В данной работе, кортежи извлекаются с помощью языковых моделей, включая авторегрессионные, которые были использованы в заполнении графов знаний ([9]). В [10] использовалась модель GPT [11] для классификации отношения по заданным двум сущностям. Тем не менее, в данной работе авторегрессионная модель работает в обратном режиме, то есть по заданной реплике и отношению, она генерирует две сущности находящиеся в реплике в данном отношении.

Извлечение информации В обработке естественного языка очень важны и хорошо изучены подходы извлечения информации в открытой и закрытой форме ([12], [13], [14], [15]). Научным сообществом были представлены методы основанные, и на шаблонах ([16], [17]), и на обучаемых моделях ([18], [19], [20], [21]), однако большинство из этих подходов извлекают информацию путем проставления тегов на части предложения. Дополнительно можно считать, что задача поставленная в этой работе относится к семейству задач отслеживания состояния диалога (dialogue state tracking [22]).

2 Данные

Датасета для обучения моделей для извлечения атрибутов пока не существует. Из-за этого, в данной работе взаимствован датасет GTKY из статьи [1]. Он создан методом distant supervision из датасета DialogueNLI [23], который в свою очередь построен на основе датасета PersonaChat [5]. Ниже будет дан краткий обзор каждого из датасетов.

PersonaChat Датасет содержащий примерно 10.000 диалогов открытого домена, в среднем состоящие из 10-15 реплик. Имеется разметкая каждого диалога по персонам собеседников. Пример диалога можно увидеть на Рисунке 2. В датасете насчитывается 1155 персон с 5000 предложениями в неструктурированном формате на естественном языке, которые идут перед диалогами, но при этом, нет четкого соответствия между предложениями персоны и репликами.

DialogueNLI Относительно новый датасет для NLI задач ([24]) в диалоговом домене, построенный над датасетом PersonaChat. Состоит из пар реплик с метками импликации, нейтральности и противоречия (entailment, neutral, contradiction). Например, на Рисунке 2 реплика Персоны A ”I just got back from the club.” является следствием предложения ”I like to dance at the club.”. Каждое предложение персоны в датасете имеет разметку в виде триплета *(subject, relation, object)*, в котором *relation* - это отношение из предопределенного множества отношений, например *live_in_general*, *like_goto*, *have_pet*. В датасете DialogueNLI таких отношений 61, и все множество этих отношений можно найти в приложениях к статье [23]. С примерной структурой датасета можно ознакомиться на Рисунке 3.

GTKY Так как в датасете DialogueNLI не все реплики имеют разметку по триплетам, авторы статьи [1] применили SOTA NLI модель, чтобы соотнести предложения персон к репликам. То есть, если предложение персоны и реплика имеют положительный entailment, то какой то атрибут из этой реплики представляется в виде триплета соответствующего этому предложению персоны. Например, реплика ”I prefer basketbal; team sports are fun.” и предложение ”I like playing basketball.” имеют положительный entailment, поэтому данной реплике можно присвоить триплет персоны *(I, like_sports, basketball)* в качестве одного из атрибутов. В качестве NLI модели для определения степени entailment’а двух предложений был использован BERT [25], дообученный на датасете DialogueNLI, который получил точность 88.43% на тестовой выборке.

2.1 Анализ датасета GTKY

Размеры обучающей, тестовой и валидационной выборки равны 131.424, 15.008 и 15.586 соответственно. Со статистиками по распределению количества триплетов по репликам можно ознакомиться в Таблице 1 и на Рисунке 4. Как можно увидеть, в датасете около 50% примеров имеют пустой целевой набор атрибутов. Так же можно посмотреть на распределение типов отношений на Рисунке 5, а самые часто и самые редко встречающиеся сущности показаны в Таблице 2 и 3.

Техника distant supervision заменяет человеческую разметку данных, однако, вводит другие трудности, представленные шумом в данных что видно в Таблице 4.

Persona A	Persona B
I just bought a brand new house. I like to dance at the club. I run a dog obedience school. I have a big sweet tooth. I like taking and posting selkies.	I love to meet new people. I have a turtle named Timothy. My favorite sport is the ultimate frisbee. My parents are living in Bora. Autumn is my favorite season.
Conversation [A] Hi, I just got back from the club. [B] Cool, this is my favorite time of the year season wise. [A] I would rather eat chocolate cake during this season. [B] What club did you go to? Me and Timothy watched TV. [A] I went to club Chino. What show are you watching? [B] We watched a show about animals like him. [A] I love those shows. I am really craving cake. [B] Why does that matter any? I went outdoors to play frisbee [A] It matters because I have a sweet tooth.	

Рис. 2: Пример диалога из датасета PersonaChat. Персоны собеседников заданы перед диалогом.

	TRAIN	TEST	VALIDATION
max	6	5	5
mean	0.61	0.61	0.63
median	0	0	1

Таблица 1: Статистики количества триплетов в репликах в датасете GTKY.

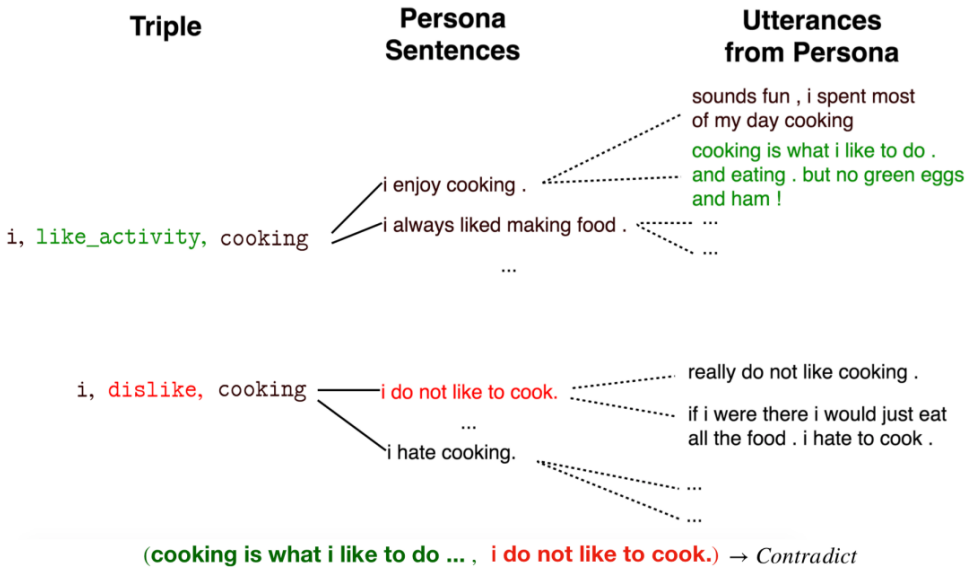

Рис. 3: Пример из датасета DialogueNLI и схема соотношения триплетов к репликам.

Субъект	Количество
i	75202
my mother	988
my	567
my brother	204
my parents	203
my career	1
my marriage	1
commitment	1
my foster	1
dude	1

Таблица 2: Топ-5 самых часто и редко встречаемых субъектов среди триплетов в обучающей выборке датасета GTKY.

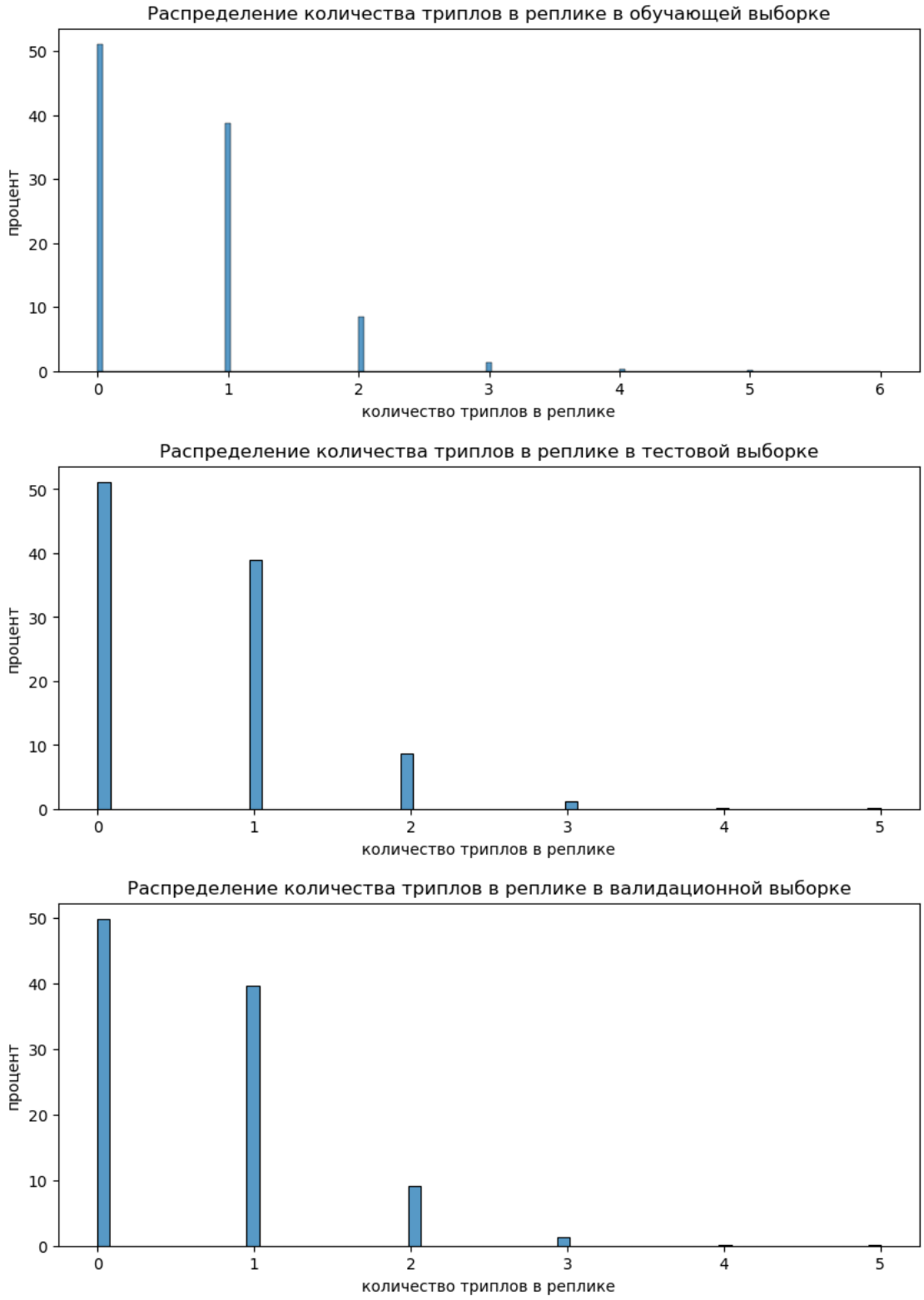

Рис. 4: Распределения количества триплетов в репликах в разных выборках датасета GTKY.

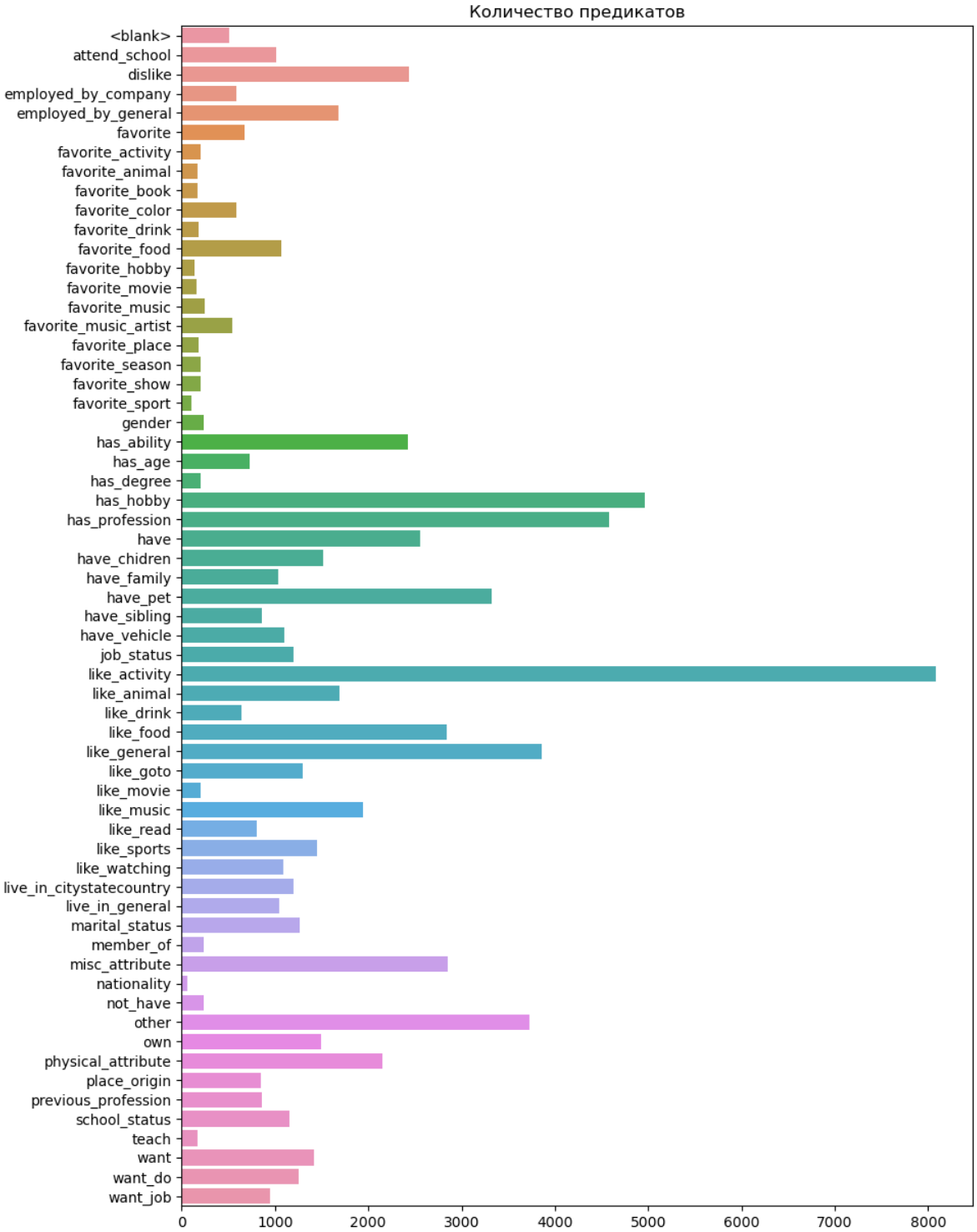

Рис. 5: Распределение типов отношений в обучающей выборке датасета GTKY.

Объект	Количество
dog	1530
<blank>	1336
cat	992
student	760
cooking	723
hard time	1
5 pets	1
carriage	1
group	1
significant other	1

Таблица 3: Топ-5 самых часто и редко встречаемых объектов среди триплетов в обучающей выборке датасета GTKY.

Реплика	триплет
all i can think about is moving away.	*(my, like_goto, desert)*
i have a wife i am in my early 40s.	*()*
she is a golden retriever very nice dogs.	*(my dog, other, name wonwon)*
i like to meet new people.	*(i, live_in_citystatecountry, new york)*
i already mentioned them. but i like to spend time with my family and dogs.	*()*
do you work anywhere ? i have a part time job.	*()*
sounds like some awesome films.	*(i, like_goto, movie theater)*
rather get to know you i got free hp computer today.	*(my adopted father, employed_by_company, hp)*

Таблица 4: Шум в разметке датасета GTKY с distant supervision.

3 Методология

Для решения поставленной задачи предлагаются два основных подхода: прямой sequence-to-sequence подход (GenAE) и двухэтапный метод (PipelineAE). В данном разделе описываются эти подходы и их достоинства с недостатками.

3.1 GenAE

С общей архитектурой этого метода можно ознакомиться на Рисунке 6. В данном подходе используется sequence-to-sequence модель для генерации набора триплетов непосредственно из реплики. То есть решается задача языкового моделирования [26]:

$$\log P(y_i|y_{i-1}, ..., y_0, \mathbf{X}) \rightarrow \max, \tag{1}$$

где y_i - это:

- правильный следующий токен из словаря модели;
- токен `<subj>` - специальный токен, после которого должен сгенерироваться субъект в триплете;
- токен `<rel>` - специальный токен, после которого должно сгенерироваться отношение в триплете;
- токен `<obj>` - специальный токен, после которого должен сгенерироваться объект в триплете;
- токен `<none>` - специальный токен, который означает, что реплика не содержит триплетов;

и $\mathbf{Y} = (y_0, ..., y_i, ..., y_n)$ - это набор триплетов разеделнных знаком ";" либо "`<none>`", а $\mathbf{X}$ - это входная реплика. То есть модель обучается на максимизации логарифма вероятности следующего токена учитывая контекст и вход. Следует заметить, что подходит и обычная генеративная модель, то есть нет преимущества использовать обязательно encoder-decoder модели. Но в экспериментах в данной работе были использованы mT5 в base и small размерах, которые имеют архитектуру encoder-decoder.

Например, по реплике "My brother likes to eat plov." модель должна сгенерировать "`<subj> I <rel> have_sibling <obj> brother; <subj> my brother <rel> like_food <obj> plov`", а по реплике "Hello, how are you doing today?" - "`<none>`".

Основным преимуществом GenAE является то, что он прост в реализации и хорошо работает в случаях, когда реплики короткие и содержат не более одного триплета. Так как в остальных случаях, для лучшего качества требуется соблюсти порядок триплетов в котором они встречаются во входной реплике. Важно так же заметить, что результаты модели плохо интерпретируемы, так как нет возможности узнать уверенность модели в выборе определенного отношения из общего множества. Брать уверенность модели в сгенерированных токенах отношения во время генерации некорректно из-за того, что вероятность в таком случае вычисляется не по множеству отношений, а по множество всех токенов в словаре модели. Другой недостаток модели в том, что генеративные модели не предобучаются извлекать структурированную информацию из входного текста, и применение их в подобных задачах может показать низкие результаты, если неправильно аугментировать их способности. Например, в данной задаче на одну модель накладывается несколько подзадач одновременно: определение релевантных отношений в реплике и генерация сущностей. Это может помешать модели выделять сразу несколько триплетов из реплик, что и наблюдалось в экспериментах.

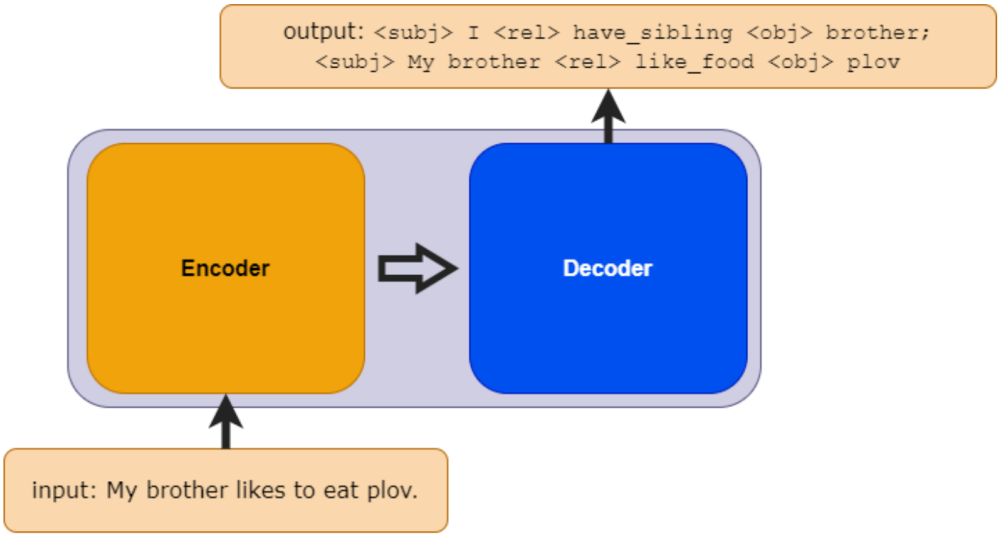

Рис. 6: Общая архитектура решения прямым методом используя seq2seq модель.

3.2 PipelineAE

Общая архитектура данного подхода изображена на Рисунке 7. В нем поставленная задача делится на две очевидные подзадачи: выявление существующих в реплике отношений и по этим отношениям и реплике генерация сущностей: субъектов и объектов. Ниже будут описаны модели и методы, которые решают эти подзадачи. Из достоинств PipelineAE можно выделить интерпретируемость результатов и расширяемость, так как, как можно бу-

дет узнать позже, большинство методов решения этих подзадач применимы в few-shot подходах. Так же модели обучаются одновременно благодаря тому, что данные на которых они обучаются независимы. Из недостатков можно отметить увеличенный объем вычислений и моделей, что может затруднять поддержку этих моделей в индустриальном применении.

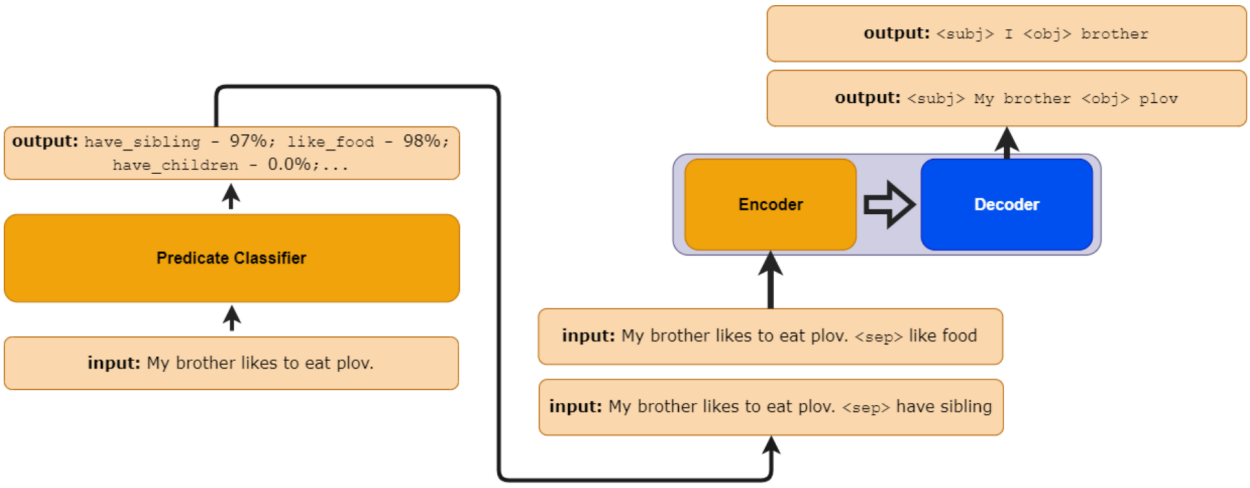

Рис. 7: Общая архитектура PipelineAE. Реплика сначала подается на вход классификатору предикатов (слева) и по выделенным предикатам и реплике генерируются сущности (справа).

3.2.1 Классификатор предикатов

Задачу выявления отношений из реплик можно сформулировать как multi-label classification, то есть по входной реплике предсказывается множество отношений которые присутствуют в ней, как на Рисунке 8. Стоит заметить, что множество может быть и пустым.

Multi-label classification Можно решать задачу напрямую и в таком случае телом классификатора может служить любая SOTA BERT-подобная модель для текстовой классификации, эмбеддинг токена `[CLS]` которой в последнем слое проходит через линейный слой и активацию vector-wise sigmoid:

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

В качестве функции потерь тут служит binary cross-entropy loss:

$$L(x, y) = \frac{1}{C} \sum_{i=1}^{C} l_i(x, y), \tag{3}$$

где C - количество классов, в данном случае количество предопределенных отношений $|R|$,

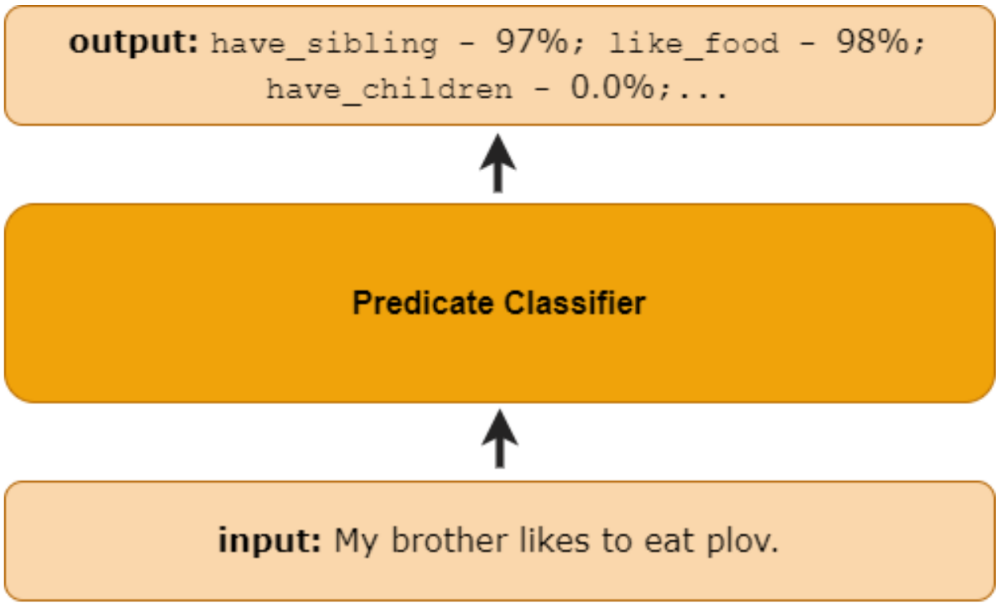

Рис. 8: Классификатор предикатов по входной реплике. Результат представляется в режиме multi-label classification.

$$R = \{r_1, ..., r_C\} \tag{4}$$

, x - logit'ы модели по входной реплике,

$$y = ([r_1 \in Y], [r_2 \in Y], ..., [r_C \in Y])^\intercal \tag{5}$$

, Y - множество отношений в реплике, а

$$l_i(x, y) = -[w_i \cdot y_i \cdot \log p_i + (1 - y_i) \cdot \log (1 - p_i)], \tag{6}$$

где $p_i = Sigmoid(x_i)$. В формуле 6 можно увидеть w_i - вес присутсвия i-го предиката в целевом множестве отношений по данному входу. Например, если в датасете у 100 реплик есть этот предикат в целевых наборах предикатов, а у 300 его нет, то $w_i = \frac{300}{100} = 3$, что значит, что для модели в датасете не будет дизбаланса классов, что полезно в случаях multi-label classification когда датасет всегда дизбалансный, если рассматривать маргинальные распределения каждого предиката по отдельности. Этот метод широко известен под названием binary-relevance [27] и в нем каждый класс рассматривается независимо от других, что не всегда может быть правильно в некоторых задачах, если классы имеют некоторую иерархическую онтологию.

Binary ranking В таком подходе ставится задача определения релевантности отношения к реплике, то есть по входной реплике и описанию предиката ставится задача бинар-

ной классификации. В качестве модели можно снова брать любую SOTA BERT-подобную модель для классификации пар текстов. Можно подавать вместо описания предиката сам предикат, но тогда качество будет относительно хуже, из-за небольшого количества семантической информации и присутствия токена ”_” нижнего подчеркивания в отношении, который непривычен для предобученных языковых моделей. Таким, образом, на вход модели передается строка в формате ”X `[SEP]` Y”, где X - реплика, а Y - описание предиката; и на выходе модели ожидается вероятность того, находится ли предикат соответствующий описанию Y в реплике X. Для такого подхода датасет преобразовывается следующим образом:

1. рассматривается каждый пример из исходного датасета GTKY;

2. если у реплики непустой целевой набор атрибутов, то для каждого отношения из этого набора формируется ”положительная” пара ”X `[SEP]` Y” и добавляется в новый датасет, а из дополнения этого набора от общего множества случайным образом добирается N отношений чтобы сформировать ”отрицательные” пары. Таким образом, каждой положительной паре соответствует N отрицательных пар;

3. если у реплики пустой целевой набор атрибутов, то к этой реплике из множества всех отношений добирается N / 2 отношений для формирования отрицательных пар и добавляется в новый датасет. Эвристика деления N на 2 помогает избавиться от чрезмерного увеличения отрицательных примеров в новом датасете, так как, как можно было увидеть ранее, исходный датасет содержит очень много реплик без каких-либо атрибутов.

Этот подход называется negative sampling и часто используется для обучения моделей retrieve’а документов или в построении графов знаний. От параметра N зависит размер полученного датасета и отношение положительных и отрицательных примеров. Если N слишком большой, то будет смещение в сторону отрицательного класса, а если N слишком маленький, то модель не научится различать все пары реплик и предикатов. В экспериментах были рассмотрены N=4 и N=9, и было замечено что N=4 дает более приемлимый результат. Важно так же заметить, что при отборе неподходящих отношений нужно рассматривать их равновероятно и с разными seed’ами, чтобы в полученном датасете каждый тип отношений встречался примерно одинаковое количество раз, тогда модель сможет увидеть все отношения из множества и оценить их релевантность к репликам. В качестве функции потерь берется аналогичная 6 функция, однако не учитывается вес положительных примеров, так как баланс классов контролируется параметром N.

Для предсказания по входной реплике X набора отношений с помощью такой модели, в нее подается каждая пара "X [SEP] $desc(r_i)$" $\forall r_i \in R$ и получаются оценки релевантности, и оставляются только те отношения, для которых оценка выше определенного порога. Очевидно, таким образом, сложность процесса предсказания модели равна $O(M \cdot C)$, где M - количество реплик в тестовой выборке и C - количество отношений, в то время как у предыдущего подхода она равна $O(M)$. Это основной недостаток данного подхода, однако это дает ему возможность легко обобщаться на новые предикаты, благодаря рассмотрению их описаний на естественном языке.

Contrastive learning В этом подходе также используется negative sampling: для каждой реплики и каждого положительного предиката добирается по N отрицательных предикатов. Для случаев, когда реплика не имеет предикатов, дополнительно вводится предикат <none> и добавляется в множество отношений, то есть он может быть отобран в процессе negative sampling.

После вышеуказанного преобразования исходного датасета, можно предположить, что получится датасет:

$$\mathcal{D} = \{(q_i, r_i^+, r_{i,1}^-, .., r_{i,N}^-)\}_{i=1}^M, \tag{7}$$

который состоит из M примеров, каждый из которых содержит реплику q_i , один релевантный (положительный) предикат r_i^+ и N отрицательных предикатов. Во время обучения минимизируется отрицание логарифма правдоподобия положительного предиката:

$$L(q_i, r_i^+, r_{i,1}^-, .., r_{i,N}^-) = -\log \frac{e^{sim(q_i, r_i^+)}}{e^{sim(q_i, r_i^+)} + \sum_{j=1}^N e^{sim(q_i, r_{i,j}^i)}}, \tag{8}$$

где,

$$sim(q, r) = BERT(\text{"}q\,\texttt{[SEP]}\,desc(r)\text{"}), \tag{9}$$

то есть используется архитектура BERT-подобной модели для задачи регрессии, эмбеддинг токена [CLS] который проходит через полносвязный слой который на выходе дает одно вещественное число - similarity. Такой подход был использован в dense passage retrieval [28] и показал хороший результат.

В таком подходе, N также является важным параметром, и чем он больше, тем лучше модель учится оценивать правдоподобие релевантного предиката, но это увеличивает количество вычислений в N раз, и во время обучения произойдет $O(M \cdot N)$ forward вызовов у модели, что конечно является основным недостатком данного подхода. Тем не менее,

этот подход так же легко обобщается на новые предикаты, и часто применяется для few-shot классификации.

Предсказания можно получить аналогично предыдущему подходу, только теперь модель на выходе дает не вероятности, а неограниченные вещественнозначные числа - similarity между репликой и описанием предиката на естественном языке. Это может стать проблемой при выборе порога выбора ответа, так как сложно оценить диапазон значений, но эту проблему можно устранить функцией активации которая сжимает значения схожести в определенном отрезке, но это может вызвать проблему затухающих градиентов.

3.2.2 Генератор сущностей

После определения всех предикатов в реплике, ставится задача генерации сущностей по заданной реплике и отношению, которые находятся в данном отношении. Можно свести данную задачу к задаче заполнения недостающих частей текста (*span corruption*) чем хорошо справляются современные sequence-to-sequence модели. Задачу можно продемонстрировать на Рисунке 9, либо таким примером:

```
utt = i have two boys and they are always hungry.
rel = have children
input:
{utt} <sep> <subj> <span_1> <rel> {rel} <obj> <span_2>
output:
<span_1> i <span_2> 2 son
```

, где `<span_1>` и `<span_2>` обозначают пропуски в тексте. В данном случае пропусков всегда два: это субъект и объект триплета которые нужно сгенерировать. По сути, задача такая же как в GenAE 3.1, и оптимизируется такая же функция потерь как в 1, но вход в модель теперь содержит и существующее отношение в реплике. Важно заметить, что модель в таком подходе обучается генерировать сущности только по отношению которое присутствует в реплике, и если подавать отсутствующий в реплике предикат, поведение модели неопределено.

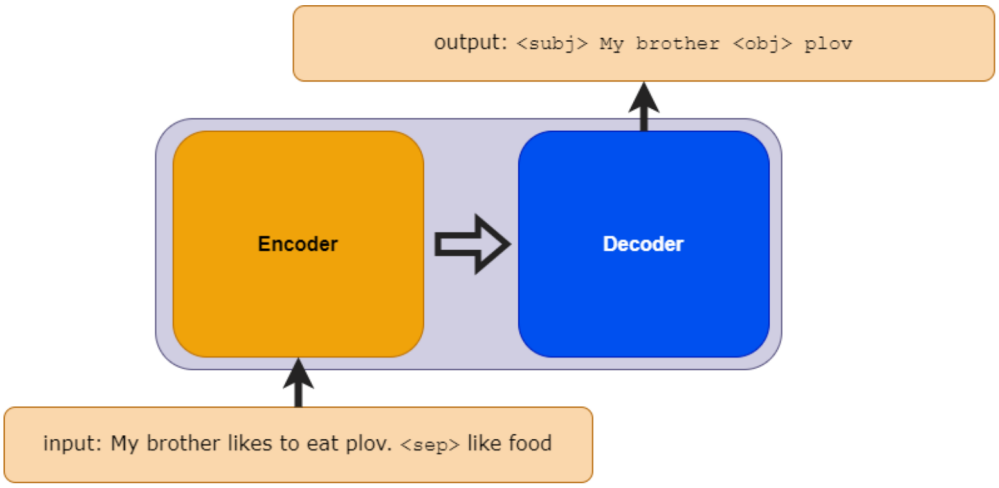

Рис. 9: Генератор сущностей использующий seq2seq модель. Принимает на вход пару: реплика и предикат, который присутствует в данной реплике, и генерирует субъект и объект находящиеся в соответствующем отношении в реплике.

4 Эксперименты

В данном разделе описаны детали экспериментов проведенных используя вышеописанную методологию и метрики качества вместе с базовыми моделями. Тут важно отметить, что в экспериментах были использованы многоязычные версии популярных трансформеров, чтобы их можно было дообучить с небольшим количеством данных на целевом языке и получить рабочие модели на этом языке.

4.1 Детали обучения

Все модели обучались на двух видеокартах NVIDIA A100 40gb. В качестве оптимизатора был использован AdamW [29] с `weight_decay=0.01`. Шаги обучения и размеры batch'а варьировались в зависимости от модели. Ниже описаны детали по каждому подходу (размеры batch'а указаны для каждого вычислительного устройства):

GenAE В качестве sequence-to-sequence моделей были выбраны mT5-small и mT5-base [30]. Шаг обучения для обеих моделей был равен 0.0003, а размер batch'а для базового варианта модели 32, а для маленького варианта - 64. Обучение длилось 3 эпохи с классом `Seq2SeqTrainer` библиотеки Huggingface Transformers [31]. Генерация происходит жадным декодированием, то есть в качестве следующего токена всегда выбирается токен с наибольшей условной вероятностью.

Классификатор предикатов В качестве базовой архитектуры BERT-подобной модели для текстовой классификации были выбраны многоязычный DistilledBERT [32] и mBERT-base. Для multilabel classification были подобраны веса положительных классов как описано в формуле 6 в одном эксперименте, и в другом - приравнивались 1. Шаг обучения для всех экспериментов с этой моделью был равен 0.00003, а для binary ranking и contrastive learning 0.00002. Размер batch'а для multilabel classification был равен 512, для contrastive learning - 10, для binary ranking - 128. Для binary ranking были так же рассмотрены mBERT-base и mDeBERTa-NLI base. NLI модель рассматривалась для улучшения качества определения релевантности предиката. Изначально, в экспериментах предикаты подавались на вход моделям как есть, позже было решено использовать их описания на естественном языке, что везде улучшило качество. Во всех подходах модели обучались в течение 5 эпох, кроме contrastive learning в котором обучение длилось 4 эпохи.

Процессы получения предсказаний у подходов binary ranking и contrastive learning очень похожи, то есть обе модели принимают на вход пару реплики и описания предиката и дают на выходе число, либо вероятность релевантности предиката, либо схожесть описания

предиката и реплики, и для корректности предсказаний требуют установления определенного порога. Порог можно подобрать жадным образом. Для простоты, в данной работе порог был подобран для всех предикатов один в обоих подходах. Это также связано с предположением, что в данных подходах модели не совсем ограничены предопределенным множеством предикатов и обучаются на описаниях отношений на естественном языке, а не на самих предикатах, для достижения обобщающей способности. Таким образом:

1. для binary ranking выбирается равномерная сетка в отрезке $[0, 1]$ и считается multi-label micro-averaged F1 на валидационной выборке и выбирается порог с наибольшим показателем данной метрики;

2. для constrastive learning действуем аналогично пункту 1, но равномерная сетка строится на отрезке $[\min, \max]$, где min - минимальное значение схожести на валидационной выборке, а max - соответственно наибольшее;

3. если же в contrastive learning модель на выходе дает числа из специфичного диапазона, например из $[-1, 1]$ или $[0, 1]$ поступаем аналогично пункту 1.

Генератор сущностей Для этого подхода был использован mT5-small. В качестве оптимизатора был изначально выбран AdaFactor [33], с помощью которого все T5 модели были предобучены, однако независимо от размера шага, с этим оптимизатором генератор сущностей переобучался начиная уже с ранних этапов обучения. Из-за этого по итогу было решено обучать только оптимизатором AdamW с `lr=0.00005` с размером batch'а 64. Дополнительно, для ускорения обучения модель была обучена в mixed-precision режиме с float16, но по итогу эксперимента во всех случаях было переполнение в представлении вещественных чисел float16. В каждом подходе модели обучались в течение 5 эпох. Генерация как и в GenAE, происходит жадным декодированием.

Метрики качества Чтобы оценить итоговое качество извлечения триплетов из реплик берутся accuracy, F1 и BLEU-1. Accuracy и F1 считаются по точному совпадению полученных триплетов: recall равен отношению количества правильно сгенерированных триплетов на количество всех триплетов в тестовой выборке, а precision - это отношение количества правильно сгенерированных триплетов к количеству всех сгенерированных триплетов. В то время, как BLEU-1 более гибкий и учитывает нестрогое совпадение между триплетами. Для оценки качества классификатора предикатов применялись accuracy и multi-label F1. А генератор сущности оценивался в конечном режиме, когда в качестве извлеченных предикатов подавались правильные отношения из датасета.

GenAE на WikiNRE Для оценки качества sequence-to-sequence подхода был проведен дополнительный эксперимент на датасете WikiNRE [34]. Детали эксперимента и результаты описаны в Приложении А.

5 Результаты

В данном разделе будут описаны результаты экспериментов и сравнения разных подходов, так же сделаны некоторые выводы. Все показатели моделей представленных в данной работе отобраны из лучших checkpoint'ов. Стоит отметить, что показатели даны в качестве справки, так как в данном исследовании основной целью было построить рабочий framework и протестировать его на доступном датасете. Так как сам датасет очень шумный из-за разметки с использованием distant supervision, результаты моделей недостаточно высоки.

5.1 Базовая модель

В качестве лучшего существующего подхода для извлечения атрибутов из реплик была выбрана модель GKTY представленная в статье [1]. Она выбрана в качестве базовой модели, так как из всех существующих похожих подходов она решает поставленную задачу как есть, без дополнительных условностей. Модель состоит из трех компонент:

- encoder входной реплики (bidirectional GRU [35]) для получения эмбеддингов;
- классификатор предикатов: multi-hop end-to-end memory network [36] которая хорошо показала себя в задаче ответов на вопросы. Решает задачу multi-label classification;
- генератор сущностей (GRU [35]) который по скрытому состоянию encoder'а и предсказанному множеству предикатов генерирует соответствующие субъект и объект.

Стоит заметить, что все три компоненты этой архитектуры обучаются вместе в сквозном режиме. В PipelineAE каждая компонента может обучаться параллельно. Также можно заметить, что GenAE тоже напоминает базовую модель, так как одна архитектура решает сразу несколько задач, но все происходит неявно во время оптимизации одной функции потерь 1, в то время как у базовой модели используется взвешенная сумма разных функций потерь под каждую задачу.

5.2 Классификатор предикатов

В PipelineAE требуется оценить качество каждой компоненты по отдельности. В Таблице 5 даны оценки качества классификации предикатов. Предсказания каждого подхода непосредственно или с некоторым преобразованием сводятся к multi-label classification, поэтому даны показатели метрик описанных в разделе экспериментов для multi-label классификации. Для binary ranking и contrastive learning даны результаты после подбора оптимального

порога на валидационной выборке. Можно увидеть, что применение multi-label classification без подбора весов для положительных классов сильно смещается в сторону пустого целевого набора предикатов. Об этом говорят нулевая точность (precision) и полнота (recall), то есть модель не дает на выходе 1. Благодаря этому, accuracy этой модели равно доле реплик в датасете в которых нет предикатов вообще. После взвешивания положительных классов, получены результаты чуть лучше, но все равно это очень низко, и модель все еще смещается в сторону 0 в каждом классе. После изучения маргинальных распределений предикатов выяснилось, что и без реплик в которых нет предикатов, в датасете по каждому классу имеется доминирование 0. Тем не менее, после подбора порога получены результаты намного лучше. В случае binary ranking NLI модель показывает самые высокие результаты, уступая по F1 только классификатору предикатов в базовой модели. Тем не менее, более тщательным подбором гиперпамараметров и преобразованием данных можно получить показатели еще выше. Перенос знаних полученных в задаче NLI очевидно в данном случае помогает улучшить результат полученный моделью предобученной на MLM и next sentence prediction. было так же замечено, что из-за своего размера mDeBERTa начала переобучаться и было решено брать checkpoint на ранних стадиях обучения. В случае контрастного обучения, результат представлен у модели, которая обучалась дополнительно с активацией $Sigmoid$ на последнем слое. Хотя, независимо от того, есть сужение значений у функции или нет, модель смещается в сторону предиката "<none>". Чтобы бороться с этим феноменом можно не рассматривать все реплики с нулевым количеством предикатов, так как они очень часто похожи друг на друга, а выбрать среди них случайное подмножество, либо исследовать "сложные" примеры с помощью уверенности модели. Тем не менее, это исследование довольно трудоемко и оставляется на будущее. После подбора порога было замечено, что для достижения оптимального multi-label F1 у моделей в обоих подходах выбирается очень высокий порог. Для binary ranking ≈ 0.87, а для contrastive learning с активацией $Sigmoid \approx 0.98$. Возможно, это также связано с маргинальными распределениями каждого из предикатов в датасете.

5.3 Генератор сущностей

В Таблице 6 представлены результаты измерения качества генератора сущностей. Без особых приемов и техник кроме использования описаний предикатов вместо них самих дает достигается высокий результат. Эти показатели можно брать за потолок, который можно достичь с классификатором предикатов.

	F1	ACC
mBERT MC weighted	14.09	17.11
mBERT MC weighted (с подбором порога)	30.03	41.4
mBERT MC	0.0	52.04
DistilBERT CL	28.38	26.39
DistilBERT BR	33.39	42.86
mDeBERTa NLI BR	38.9	**45.64**
Базовый классификатор предикатов	**44.40**	41.57

Таблица 5: Результаты качества классификации предикатов. BR - binary ranking, CL - contrastive learning, MC - multi-label classification

	ACC	F1	BLEU-1
mT5-small	**70.3**	**70.6**	**96.55**
Базовый генератор сущностей	43.48	46.03	-

Таблица 6: Результаты качества генерации сущностей по заданной реплике и правильному предикату.

5.4 Итоговые метрики

В Таблице 7 даны показатели оценок итоговых моделей. Как можно увидеть, почти все подходы представленные в данной работе оказались лучше чем модель GTKY, что показывает превосходство трансформерной архитектуры в данной задаче. Контрастное обучение не дало удовлетворительного результата, что требует дальнейшего более подробного исследования и анализа. Самый лучший результат у модели с классификатором предикатов NLI binary ranking. Следует заметить, что всех моделях в двух-этапном подходе меняется только классификатор предикатов, а генератор сущностей один и тот же. Важно заметить, что показатели генератора сущностей сильно занижаются из-за качества классификации предикатов.

На Рисунке 10 изображены распределения количества разных сгенерированных триплетов по репликам у различных подходов. Можно увидеть, что mT5 очень редко генерируют больше одного триплета. Только взяв модель базового размера можно было наблюдать заметный скачок в качестве и в распределении предсказанных триплетов. Так же, было замечено, что часто mT5 генерирует один и тот же триплет дважды либо даже 12 раз.

Это, возможно, следствие жадного декодинга который использовался во всех экспериментах с генерацией, либо неконсистентный порядок в наборе целевых триплетов, что мешает авторегресссионной модели усваивать правильную последовательность триплетов если их несколько. Распределение триплетов предсказанных GenAE больше всех похоже на распределение в тестовой выборке на Рисунке 4.

	ACC	**F1**	**BLEU-1**
PipelineAE с mBERT-base MC weighted	39.17	38.9	68.68
PipelineAE с DistilBERT BR	35.64	38.06	68.49
PipelineAE с DistilBERT CL	25.8	26.73	66.39
PipelineAE с mDeBERTa NLI BR	40.49	43.13	70.84
GenAE с mT5-small	40.25	41.28	71.92
GenAE с mT5-base	**45.28**	**46.11**	**72.84**
Базовая модель	26.52	28.68	51.87

Таблица 7: Итоговые оценки качества моделей для извлечения триплетов из реплик. Показатели получены с подобранным порогом для классификатора предикатов.

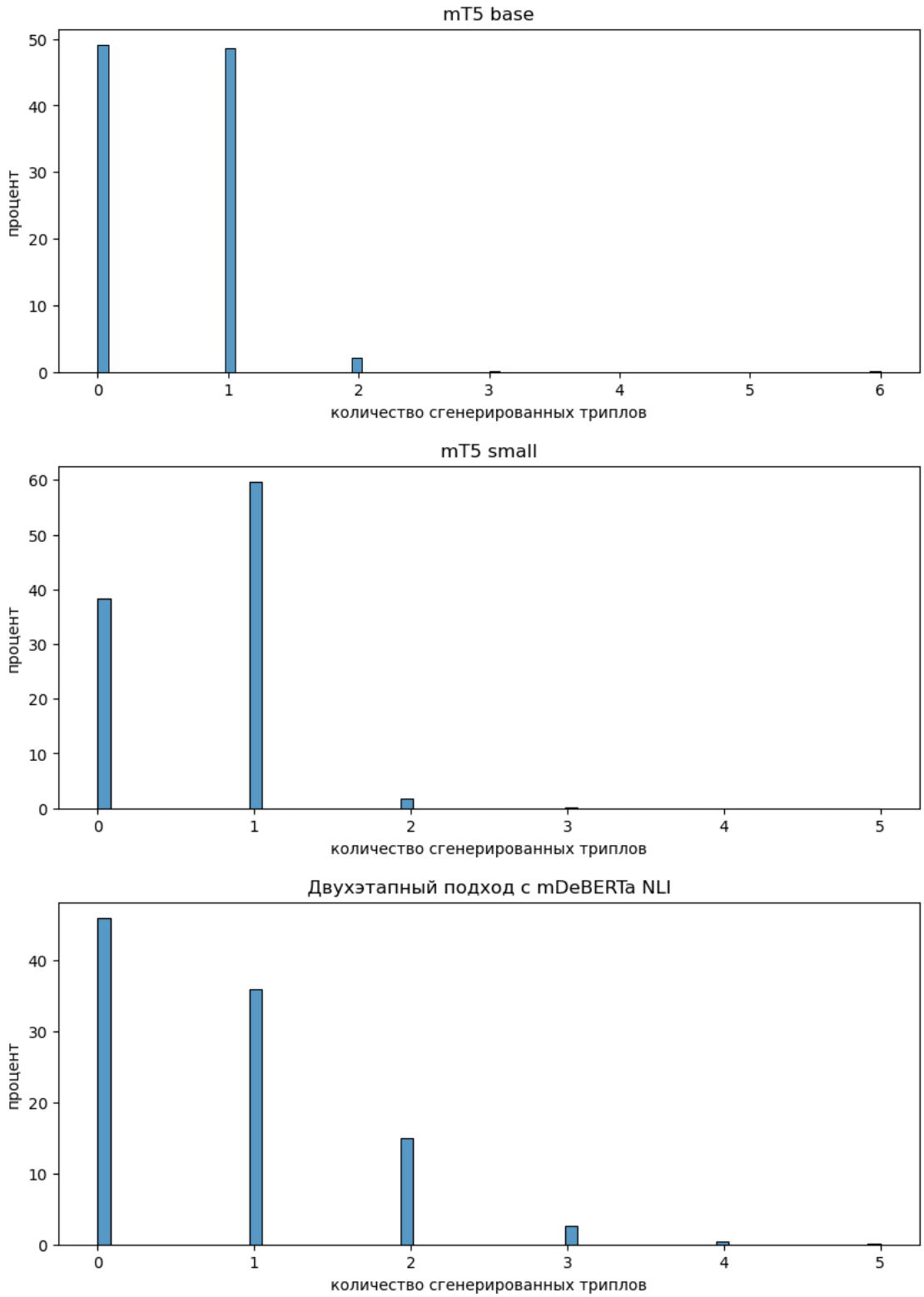

Рис. 10: Распределения количества триплетов в предсказаниях моделей.

6 Применения

У извлечения пользовательских атрибутов из диалогов в структурированном формате имеется потенциально очень много приложений. Например, можно использовать в поисковиках, в рекомендательных системах друзей, в социальных науках, в персонализованных диалоговых ассистентах и так далее. Причинами проведения данной работы послужили следующие применения: персонализованных диалоговые ассистенты и рекомендательные системы.

Персонализованные диалоговые ассистенты Огромное внимание уделяется разработке систем, которые могут повысить качество беседы и сделать ее более увлекательной для участников. Есть два направления развития агентов персонализированного диалога: придание агентам индивидуальности как в статье про PersonaChat [5], или адаптация агентов к их конечным пользователям с помощью пользовательских атрибутов. Следовательно, если можно снабдить диалоговую систему модулем извлечения пользовательских атрибутов, это станет шагом в развитии персонализированных диалоговых систем.

Диалоговая система может рассматривать пользовательские атрибуты, извлеченные из истории диалога, как долгосрочную память. Эта информация позволяет избежать системе повторять одни и те же или похожие вопросы. Например, если пользователь упомянул, что он родился в сентябре 2009 года в предыдущем разговоре два дня назад, система персонализированного диалога должна избегать задавать вопросы типа ”В каком месяце у вас день рождения?” или ”А сколько тебе лет?”. Кроме того, такие атрибуты можно использовать для фильтрации ответов системы или для предложения ответов получше. Например, для персонализированной системы было бы неуместно спрашивать ”Как у вас дела в универе?”, если пользователь родился в 2013м году, в то время как сейчас 2023й год. Лучше будет, если система ответит ”Вау! Скоро тебе исполнится 10!” после получения информации о времени рождения пользователя.

Персонализованные рекомендательные системы Существует три основных типа рекомендательных систем: knowledge-based система содержит атрибуты пользователя и продуктов и дает рекомендации на основе сходства этих атрибутов; content-based система рекомендует товары, похожие тем, которые понравились данному пользователю в прошлом, независимо от предпочтений других пользователей; в то время, как система на основе collaborative-filtering делает рекомендации основываясь на прошлых взаимодействиях всей базы пользователей, например, рассматривая k-ближайших соседей данного пользователя.

Большинство этих рекомендательных систем требуют фактического взаимодействия

пользователей с продуктами, такими как щелчок мышью или просмотр. Используя решения из данной работы можно собирать пользовательские атрибуты в неявном режиме, которые затем можно использовать в кластеризации пользователей или для сохранения продуктов упомянутые пользователем в прошлом. Например, если имеются два пользователя и оба из Москвы и любят хоккей, то можно порекомендовать игру ЦСКА одному пользователю, если другой часто упоминает ее.

ЗАКЛЮЧЕНИЕ

В данной работе было исследовано извлечение атрибутов пользователей из разговорных данных. Из-за отсутствия датасета с качественной разметкой была использована техника distant supervision, которая добавила значительный шум в данные. Тем не менее, почти все методы из этой работы получили результат лучше чем базовая модель, показавшая в использованном датасете самые высокие показатели. Так же были описаны достоинства и недостатки каждого из предложенных подходов, а так же потенциальные применения этих методов. Так же были помечены направления для дальнейшей разработки и исследований.

Список литературы

1. Getting To Know You: User Attribute Extraction from Dialogues / C.-S. Wu [и др.] // Proceedings of the Twelfth Language Resources and Evaluation Conference. — Marseille, France : European Language Resources Association, 05.2020. — С. 581—589. — ISBN 979-10-95546-34-4. — URL: https://aclanthology.org/2020.lrec-1.73.
2. *Pappu A.*, *Rudnicky A.* Knowledge Acquisition Strategies for Goal-Oriented Dialog Systems // Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL). — Philadelphia, PA, U.S.A. : Association for Computational Linguistics, 06.2014. — С. 194—198. — DOI: 10.3115/v1/W14-4326. — URL: https://aclanthology.org/W14-4326.
3. Listening between the Lines: Learning Personal Attributes from Conversations / A. Tigunova [и др.] // CoRR. — 2019. — Т. abs/1904.10887. — arXiv: 1904.10887. — URL: http://arxiv.org/abs/1904.10887.
4. Training Millions of Personalized Dialogue Agents / P.-E. Mazaré [и др.] // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. — Brussels, Belgium : Association for Computational Linguistics, 10-11.2018. — С. 2775—2779. — DOI: 10.18653/v1/D18-1298. — URL: https://aclanthology.org/D18-1298.
5. Personalizing Dialogue Agents: I have a dog, do you have pets too? / S. Zhang [и др.] // CoRR. — 2018. — Т. abs/1801.07243. — arXiv: 1801.07243. — URL: http://arxiv.org/abs/1801.07243.
6. Call for Customized Conversation: Customized Conversation Grounding Persona and Knowledge / Y. Jang [и др.] // CoRR. — 2021. — Т. abs/2112.08619. — arXiv: 2112.08619. — URL: https://arxiv.org/abs/2112.08619.
7. Extracting and Inferring Personal Attributes from Dialogue / Z. Wang [и др.] // Proceedings of the 4th Workshop on NLP for Conversational AI. — Dublin, Ireland : Association for Computational Linguistics, 05.2022. — С. 58—69. — DOI: 10.18653/v1/2022.nlp4convai-1.6. — URL: https://aclanthology.org/2022.nlp4convai-1.6.
8. Attention is All you Need / A. Vaswani [и др.] // Advances in Neural Information Processing Systems. Т. 30 / под ред. I. Guyon [и др.]. — Curran Associates, Inc., 2017. — URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

9. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction / A. Bosselut [и др.] // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. — Florence, Italy : Association for Computational Linguistics, 07.2019. — С. 4762—4779. — DOI: 10.18653/v1/P19-1470. — URL: https://aclanthology.org/P19-1470.
10. *Alt C.*, *Hübner M.*, *Hennig L.* Improving Relation Extraction by Pre-trained Language Representations // Automated Knowledge Base Construction (AKBC). — 2019. — URL: https://openreview.net/forum?id=BJgrxbqp67.
11. Language Models are Unsupervised Multitask Learners / A. Radford [и др.]. — 2019.
12. Open Information Extraction from the Web / M. Banko [и др.] // Proceedings of the 20th International Joint Conference on Artifical Intelligence. — Hyderabad, India : Morgan Kaufmann Publishers Inc., 2007. — С. 2670—2676. — (IJCAI'07).
13. *Wu F.*, *Weld D. S.* Open Information Extraction Using Wikipedia // Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. — Uppsala, Sweden : Association for Computational Linguistics, 07.2010. — С. 118—127. — URL: https://aclanthology.org/P10-1013.
14. *Berant J.*, *Dagan I.*, *Goldberger J.* Global Learning of Typed Entailment Rules // Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. — Portland, Oregon, USA : Association for Computational Linguistics, 06.2011. — С. 610—619. — URL: https://aclanthology.org/P11-1062.
15. *Fader A.*, *Zettlemoyer L.*, *Etzioni O.* Open Question Answering over Curated and Extracted Knowledge Bases // Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. — New York, New York, USA : Association for Computing Machinery, 2014. — С. 1156—1165. — (KDD '14). — ISBN 9781450329569. — DOI: 10.1145/2623330.2623677. — URL: https://doi.org/10.1145/2623330.2623677.
16. Open Language Learning for Information Extraction / Mausam [и др.] // Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. — Jeju Island, Korea : Association for Computational Linguistics, 07.2012. — С. 523—534. — URL: https://aclanthology.org/D12-1048.

17. *Del Corro L.*, *Gemulla R.* ClausIE: Clause-Based Open Information Extraction // Proceedings of the 22nd International Conference on World Wide Web. — Rio de Janeiro, Brazil : Association for Computing Machinery, 2013. — С. 355—366. — (WWW '13). — ISBN 9781450320351. — DOI: 10.1145/2488388.2488420. — URL: https://doi.org/10.1145/2488388.2488420.

18. Relation Classification via Convolutional Deep Neural Network / D. Zeng [и др.] // Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. — Dublin, Ireland : Dublin City University, Association for Computational Linguistics, 08.2014. — С. 2335—2344. — URL: https://aclanthology.org/C14-1220.

19. Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Paths / Y. Xu [и др.] // Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. — Lisbon, Portugal : Association for Computational Linguistics, 09.2015. — С. 1785—1794. — DOI: 10.18653/v1/D15-1206. — URL: https://aclanthology.org/D15-1206.

20. Supervised Open Information Extraction / G. Stanovsky [и др.] // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). — New Orleans, Louisiana : Association for Computational Linguistics, 06.2018. — С. 885—895. — DOI: 10.18653/v1/N18-1081. — URL: https://aclanthology.org/N18-1081.

21. RESIDE: Improving Distantly-Supervised Neural Relation Extraction using Side Information / S. Vashishth [и др.] // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. — Brussels, Belgium : Association for Computational Linguistics, 10-11.2018. — С. 1257—1266. — DOI: 10.18653/v1/D18-1157. — URL: https://aclanthology.org/D18-1157.

22. *Jacqmin L.*, *Rojas Barahona L. M.*, *Favre B.* "Do you follow me?": A Survey of Recent Approaches in Dialogue State Tracking // Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue. — Edinburgh, UK : Association for Computational Linguistics, 09.2022. — С. 336—350. — URL: https://aclanthology.org/2022.sigdial-1.33.

23. Dialogue Natural Language Inference / S. Welleck [и др.] // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. — Florence, Italy : Association

for Computational Linguistics, 07.2019. — C. 3731—3741. — DOI: 10.18653/v1/P19-1363. — URL: https://aclanthology.org/P19-1363.

24. *Storks S.*, *Gao Q.*, *Chai J. Y.* Commonsense Reasoning for Natural Language Understanding: A Survey of Benchmarks, Resources, and Approaches // CoRR. — 2019. — T. abs/1904.01172. — arXiv: 1904.01172. — URL: http://arxiv.org/abs/1904.01172.

25. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin [и др.] // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). — Minneapolis, Minnesota : Association for Computational Linguistics, 06.2019. — C. 4171—4186. — DOI: 10.18653/v1/N19-1423. — URL: https://aclanthology.org/N19-1423.

26. *Bengio Y.*, *Ducharme R.*, *Vincent P.* A Neural Probabilistic Language Model // Advances in Neural Information Processing Systems. T. 13 / под ред. T. Leen, T. Dietterich, V. Tresp. — MIT Press, 2000. — URL: https://proceedings.neurips.cc/paper_files/paper/2000/file/728f206c2a01bf572b5940d7d9a8fa4c-Paper.pdf.

27. Binary relevance for multi-label learning: an overview / M.-L. Zhang [и др.] // Frontiers of Computer Science. — 2018. — Апр. — Т. 12, № 2. — С. 191—202. — ISSN 2095-2236. — DOI: 10.1007/s11704-017-7031-7. — URL: https://doi.org/10.1007/s11704-017-7031-7.

28. Dense Passage Retrieval for Open-Domain Question Answering / V. Karpukhin [и др.] // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). — Online : Association for Computational Linguistics, 11.2020. — C. 6769—6781. — DOI: 10.18653/v1/2020.emnlp-main.550. — URL: https://aclanthology.org/2020.emnlp-main.550.

29. *Loshchilov I.*, *Hutter F.* Fixing Weight Decay Regularization in Adam // CoRR. — 2017. — T. abs/1711.05101. — arXiv: 1711.05101. — URL: http://arxiv.org/abs/1711.05101.

30. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer / L. Xue [и др.] // Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. — Online : Association for Computational Linguistics, 06.2021. — C. 483—498. — DOI:

10.18653/v1/2021.naacl-main.41. — URL: https://aclanthology.org/2021.naacl-main.41.

31. HuggingFace's Transformers: State-of-the-art Natural Language Processing / T. Wolf [и др.] // CoRR. — 2019. — Т. abs/1910.03771. — arXiv: 1910.03771. — URL: http://arxiv.org/abs/1910.03771.

32. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter / V. Sanh [и др.] // CoRR. — 2019. — Т. abs/1910.01108. — arXiv: 1910.01108. — URL: http://arxiv.org/abs/1910.01108.

33. *Shazeer N.*, *Stern M.* Adafactor: Adaptive Learning Rates with Sublinear Memory Cost // CoRR. — 2018. — Т. abs/1804.04235. — arXiv: 1804.04235. — URL: http://arxiv.org/abs/1804.04235.

34. Neural Relation Extraction for Knowledge Base Enrichment / B. D. Trisedya [и др.] // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. — Florence, Italy : Association for Computational Linguistics, 07.2019. — С. 229—240. — DOI: 10.18653/v1/P19-1023. — URL: https://aclanthology.org/P19-1023.

35. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches / K. Cho [и др.] // CoRR. — 2014. — Т. abs/1409.1259. — arXiv: 1409.1259. — URL: http://arxiv.org/abs/1409.1259.

36. Weakly Supervised Memory Networks / S. Sukhbaatar [и др.] // CoRR. — 2015. — Т. abs/1503.08895. — arXiv: 1503.08895. — URL: http://arxiv.org/abs/1503.08895.

37. GenIE: Generative Information Extraction / M. Josifoski [и др.] // Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. — Seattle, United States : Association for Computational Linguistics, 07.2022. — С. 4626—4643. — DOI: 10.18653/v1/2022.naacl-main.342. — URL: https://aclanthology.org/2022.naacl-main.342.

38. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension / M. Lewis [и др.] // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. — Online : Association for Computational Linguistics, 07.2020. — С. 7871—7880. — DOI:

10.18653/v1/2020.acl-main.703. — URL: https://aclanthology.org/2020.acl-main.703.

39. *Sutskever I.*, *Vinyals O.*, *Le Q. V.* Sequence to Sequence Learning with Neural Networks // Advances in Neural Information Processing Systems. Т. 27 / под ред. Z. Ghahramani [и др.]. — Curran Associates, Inc., 2014. — URL: https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

ПРИЛОЖЕНИЕ

A Дополнительные эксперименты на датасете WikiNRE

Для оценки качества GenAE в случае консистентного порядка триплетов, был рассмотрен датасет WikiNRE собранный на основе данных из Wikipedia и графа знаний Wikidata. В качестве генеративной модели был взят T5 v1.1 small и обучен в течение 4 эпох с AdamW с `lr=0.00002` и `weight_decay=0.01` на 4 картах GeForce GTX 1080 Ti. Размер batch'а на каждое вычислительное устройство равен 16, без learning rate warmup.

Результаты представлены в Таблице 8. Для сравнения брались модели из статьи [37], которые являются наиболее популярными в построении графов знаний. Можно увидеть, что с наивным методом обучения GenAE уступает только GenIE с BART-large [38] и с constrained beam-search [39]. Тем не менее, GenAE работает лучше всех подходов основанных на нескольких моделях в виде pipeline. Результат GenAE в этом датасете намного выше, так как в нем нет шума и соблюдается порядок триплетов если их несколько во входной реплике. Тем не менее, задача поставленная в датасете схожа с основной задачей данной работы только форматом извлечения информации, а домен полностью отличается: в WikiNRE не разговорный домен и предикаты общие и не относятся к персоне определенного человека.

	Precision	Recall	F1
NeuralEL + CNN	36.89	35.21	36.03
SotA Pipeline	67.43	54.22	60.11
GenAE с T5-small	79.29	73.62	76.35
GenIE	**88.18**	**88.31**	**88.24**

Таблица 8: Результаты извлечения триплетов на датасете WikiNRE [34]. Все показатели кроме GenAE взяты из статьи [37] и [34].

B Дополнительные эксперименты на датасете DialogueNLI с ChatGPT

	Precision	Recall	F1	Accuracy	BLEU-1
ChatGPT	13.85	29.63	18.88	17.0	76.87
GenAE	**43.14**	40.74	41.9	**42.0**	76.11
PipelineAE	42.45	**54.63**	**47.77**	40.0	**83.37**

Таблица 9: Результаты извлечения триплетов на датасете DialogueNLI.